

AMENDMENT TO THE CLAIMS

Please **AMEND** claims 1, 3, 7, 16, 18, and 25 as follows.

Please **CANCEL** claims 2, 8, 9, 14, 15, 19, and 26-29 without prejudice or disclaimer.

This listing of claims will replace all prior versions, and listings, of claims in the application.

1. (currently amended) A method for optimizing performance of a database, the method comprising:

 sorting and categorizing a first set of columns within a view of the database; ~~and~~
 marking a second set of columns within the view as if the second set of columns were already sorted and categorized prior to actual sorting and categorizing of the second set of columns, the second set of columns including all columns exclusive of the first set of columns; and

sorting and categorizing at least one column of the second set of columns in response to performing a query on the at least one column,

wherein the database is a non-relational database,

the sorting and categorizing a first set of columns step includes assigning the first set of columns to a portion of a cache,

the sorting and categorizing at least one column of the second set step includes sorting and categorizing the at least one column of the second set of columns in another portion of the cache, and

the second set of columns is visible as collapsed data.

2. (canceled)

3. (currently amended) The method of claim 21, including establishing a mini-index indexing the at least one column of the second set of columns.

4. (original) The method of claim 3, including accessing the mini-index to provide increased performance.

5. (original) The method of claim 4, including monitoring parameters of the mini-index, and as a result, performing one of deleting, updating, and recreating the mini-index.

6. (original) The method of claim 5, wherein the parameters include at least one of a number of sorted columns, a number of categorized columns, a number of records that can be accessed in a view, an average number of records per category, and an average number of records per hierarchy.

7. (currently amended) A method for optimizing performance of a non-relational database, the method comprising:

 sorting and categorizing a first set of columns within a view of the non-relational database;

 marking a second set of columns within the view as if the second set of columns were already sorted and categorized prior to actual sorting and categorizing of the second set of columns, the second set of columns including all columns exclusive of the first set of columns; and

 sorting and categorizing at least one column of the second set of columns in response to performing a query on the at least one column;

maintaining the first set of columns in a portion of cache; and

maintaining the at least one column of the second set of columns in another portion of cache,

wherein the second set of columns is visible as collapsed data.

8. (canceled)

9. (canceled)

10. (original) The method of claim 7, further comprising establishing a mini-index indexing the at least one column of the second set of columns.

11. (original) The method of claim 10, including accessing the mini-index by a server to provide increased performance.

12. (original) The method of claim 11, including monitoring parameters of the mini-index, and as a result, performing one of deleting, updating, and recreating the mini-index.

13. (original) The method of claim 12, wherein the parameters include at least one of a number of sorted columns, a number of categorized columns, a number of records that can be accessed in a view, an average number of records per category, and an average number of records per hierarchy.

14. (canceled)

15. (canceled)

16. (currently amended) The method of claim ~~15~~7, further including sizing the another portion of cache depending on the size of the at least one column of the second set of columns.

17. (original) The method of claim 7, wherein the marking step permits clients to see the second set of columns and to issue a query on the at least one column of the second set of columns.

18. (currently amended) A system to optimize database performance, comprising:
a component to sort and categorize a first set of columns within a view of the database;
a component to mark a second set of columns within the view, wherein the second set of columns comprises all columns within the view that are not in the first set of columns, and wherein the mark indicates that sorting and categorizing has been performed on the second set of columns without actually having performed the sorting and the categorizing; ~~and~~
a component to sort and categorize at least one column of the second set of columns in response to a query on the at least one column; and

a component to sort and categorize the at least one column of the second set in a portion of a cache and assign the first set of columns to another portion of the cache,
wherein the database is a non-relational database, and
the second set of columns is visible as collapsed data.

19. (canceled)

20. (original) The system of claim 18, including a component to establish a mini-index that indexes the at least one column of the second set of columns.

21. (previously presented) The system of claim 20, including a component to access the mini-index by a server.

22. (previously presented) The system of claim 21, including a component to monitor parameters of the mini-index, and as a result, to perform one of deleting, updating, and recreating the mini-index.

23. (original) The system of claim 22, wherein the parameters include at least one of a number of sorted columns, a number of categorized columns, a number of records that can be accessed in a view, an average number of records per category, and an average number of records per hierarchy.

24. (original) The system of claim 18, including a component to permit clients to see the second set of columns and to issue a query on the at least one column of the second set of columns.

25. (currently amended) A computer program product comprising a computer usable medium having readable program code embodied in the medium, the computer program product includes:

a first computer program code to sort and categorize a first set of columns within a view of a database;

a second computer program code to mark a second set of columns within the view, wherein the second set of columns comprises all columns within the view that are not in the first set of columns, and wherein the mark indicates that sorting and categorizing has been performed on the second set of columns without actually having performed the sorting and the categorizing; and

a third computer program code to sort and categorize at least one column of the second set of columns in response to a query on the at least one column,

wherein the first set of columns are assigned to a portion of a cache,

the at least one column of the second set of columns is assigned to another portion of the cache,

the database is a non-relational database, and

the second set of columns is visible as collapsed data.

26. – 29. (canceled)